Technische Universität Braunschweig

Institute for Computational Mathematics

Introduction to Model Order Reduction
A Tutorial

Matthias Bollhöfer
INRIA Sophia Antipolis , July 27, 2015

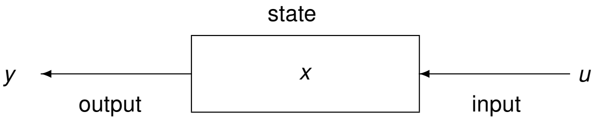
Outline

- **Model Order Reduction**
- **Proper Orthogonal Decomposition**
- **Balanced Truncation**
- **Moment Matching**
- **Conclusions**

Outline

- **Model Order Reduction**
- Proper Orthogonal Decomposition
- Balanced Truncation
- Moment Matching
- Conclusions

Objective of Model Order Reduction (MOR)



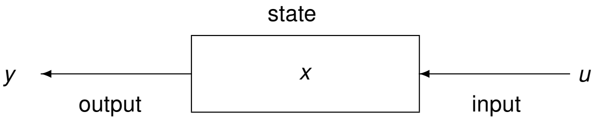
Dynamical system

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

Laplace transformation, transfer function $\mathcal{H}(s)$

$$\begin{aligned} \mathcal{H}(s) &:= C(sI - A)^{-1}B + D \\ \hat{y}(s) &= \mathcal{H}(s)\hat{u} + \dots \end{aligned}$$

Objective of Model Order Reduction (MOR)



Dynamical system

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

Laplace transformation, transfer function $\mathcal{H}(s)$

$$\begin{aligned} \mathcal{H}(s) &:= C(sI - A)^{-1}B + D \\ \hat{y}(s) &= \mathcal{H}(s)\hat{u} + \dots \end{aligned}$$

Here most of the time $B, C^* \in \mathbb{R}^n$, single-input single-output case

Objective of Model Order Reduction (MOR)

Dynamical system

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

Model Order Reduction: find full rank $S, T \in \mathbb{R}^{n,r}$ such that $r \ll n$, $S^*T = I$ and use instead

Objective of Model Order Reduction (MOR)

Dynamical system

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

Model Order Reduction: find full rank $S, T \in \mathbb{R}^{n,r}$ such that $r \ll n$, $S^*T = I$ and use instead

$$\begin{aligned} (S^*T)\,\dot{\tilde{x}} &= (S^*AT)\,\tilde{x} + (S^*B)\,u \\ \tilde{y} &= (CT)\,\tilde{x} + Du \end{aligned}$$

$$\|y - \tilde{y}\| \text{ small}$$

Technische Universität Braunschweig

Objective of Model Order Reduction (MOR)

Dynamical system

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

Model Order Reduction: find full rank $S, T \in \mathbb{R}^{n,r}$ such that $r \ll n$, $S^* T = I$ and use instead

$$\begin{aligned} \dot{\tilde{x}} &= \hat{A}\tilde{x} + \hat{B}u \\ \tilde{y} &= \hat{C}\tilde{x} \end{aligned}$$

$$\|y - \tilde{y}\| \text{ small}$$

Objective of Model Order Reduction (MOR)

Dynamical system

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

Model Order Reduction: find full rank $S, T \in \mathbb{R}^{n,r}$ such that $r \ll n$, $S^* T = I$ and use instead

$$\begin{aligned} \dot{\tilde{x}} &= \hat{A}\tilde{x} + \hat{B}u \\ \tilde{y} &= \hat{C}\tilde{x} \end{aligned}$$

$$\|y - \tilde{y}\| \text{ small}$$

$$\mathcal{H}(s) = C(sI - A)^{-1}B + D, \quad \hat{\mathcal{H}}(s) = \hat{C}(sI - \hat{A})^{-1}\hat{B} + \hat{D}$$

$$\|\mathcal{H}(s) - \hat{\mathcal{H}}(s)\| \text{ small}$$

Example

Heat equation

$$T_t = \kappa \Delta T \text{ in } \Omega = [0,1]^2$$

Example

Heat equation

$$T_t = \kappa \Delta T \text{ in } \Omega = [0,1]^2$$

$$T(0,y,t) = p(y)u(t),\ u(t) \text{ input control,}$$

$$\frac{\partial T(x,0,t)}{\partial y} = \frac{\partial T(x,1,t)}{\partial y} = 0,\ \frac{\partial T(1,y,t)}{\partial x} = -\alpha(T(1,y,t) - T_e(y))$$

Example

Heat equation

$$T_t = \kappa \Delta T \text{ in } \Omega = [0,1]^2$$

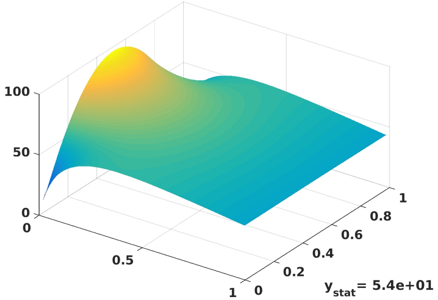
$$T(0,y,t) = p(y)u(t),\ u(t) \text{ input control},$$
$$\frac{\partial T(x,0,t)}{\partial y} = \frac{\partial T(x,1,t)}{\partial y} = 0,\ \frac{\partial T(1,y,t)}{\partial x} = -\alpha(T(1,y,t) - T_e(y))$$

$E = \int_\Omega T\, dxdy$ output, total temperature

spatial discretization (FDM/FEM)

$$\begin{aligned} \dot{T}(t) &= -AT(t) + Bu(t) \\ E(t) &= CT(t) \end{aligned}$$

A sym. pos. def. rnk$B$ refers to dependence of u w.r.t. y



Outline

- Model Order Reduction
- **Proper Orthogonal Decomposition**
- Balanced Truncation
- Moment Matching
- Conclusions

Proper Orthogonal Decomposition (POD)

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx + Du, \end{aligned} \quad \text{where } \sigma(A) \subset \mathbb{C}^-$$

Proper Orthogonal Decomposition (POD)

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx + Du, \end{aligned} \quad \text{where } \sigma(A) \subset \mathbb{C}^-$$

Compute a sequence of snapshots $\tilde{x}^{(1)} = \tilde{x}(t_1), \ldots, \tilde{x}^{(m)} = \tilde{x}(t_m)$, $\tilde{x}^{(i)} \approx x(t_i)$ f.a. i.
Set $X = \left[\tilde{x}^{(1)}, \ldots, \tilde{x}^{(m)}\right]$.

Proper Orthogonal Decomposition (POD)

$$\begin{array}{rcl} \dot{x} & = & Ax + Bu, \\ y & = & Cx + Du, \end{array} \quad \text{where } \sigma(A) \subset \mathbb{C}^-$$

Compute a sequence of snapshots $\tilde{x}^{(1)} = \tilde{x}(t_1), \ldots, \tilde{x}^{(m)} = \tilde{x}(t_m)$, $\tilde{x}^{(i)} \approx x(t_i)$ f.a. i.
Set $X = \left[\tilde{x}^{(1)}, \ldots, \tilde{x}^{(m)}\right]$.

Euclidean scalar product $(v, w) = v^* M w$, find orthonormal basis $z^{(1)}, \ldots, z^{(r)}$ s.t.

$$\sum_{i=1}^{m} \|\tilde{x}^{(i)} - \sum_{j=1}^{r} \mu_{ij} z^{(j)}\|_M^2$$

is minimized.

Proper Orthogonal Decomposition (POD)

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx + Du, \end{aligned} \quad \text{where } \sigma(A) \subset \mathbb{C}^-$$

Compute a sequence of snapshots $\tilde{x}^{(1)} = \tilde{x}(t_1), \ldots, \tilde{x}^{(m)} = \tilde{x}(t_m)$, $\tilde{x}^{(i)} \approx x(t_i)$ f.a. i.
Set $X = \left[\tilde{x}^{(1)}, \ldots, \tilde{x}^{(m)}\right]$.

Euclidean scalar product $(v, w) = v^* M w$, find orthonormal basis $z^{(1)}, \ldots, z^{(r)}$ s.t.

$$\sum_{i=1}^{m} \|\tilde{x}^{(i)} - \sum_{j=1}^{r} \mu_{ij} z^{(j)}\|_M^2$$

is minimized.

- $\mu_{ij} = (z^{(i)}, x^{(j)})$ (M-orthogonal projection)
- best rank–r approximation of X given by $M^{1/2}X = U\Sigma V^*$ (SVD), $Z_r = \left[z^{(1)}, \ldots, z^{(r)}\right] = M^{-1/2}U_r$, where U_r refers to the leading r columns of U (EMSY-Theorem).

Proper Orthogonal Decomposition (POD)

$$\tilde{x}(t) := \sum_{j=1}^{r} a_j(t) z^{(j)} \equiv Z_r a(t)$$

Proper Orthogonal Decomposition (POD)

$$\tilde{x}(t) := \sum_{j=1}^{r} a_j(t) z^{(j)} \equiv Z_r a(t)$$

Solve

$$\dot{\tilde{x}} = A\tilde{x} + Bu,\ \tilde{y} = C\tilde{x} + Du,\ \text{s.t.} \left(\dot{\tilde{x}} - A\tilde{x} - Bu\right) \perp_M Z_r$$

Proper Orthogonal Decomposition (POD)

$$\tilde{x}(t) := \sum_{j=1}^{r} a_j(t) z^{(j)} \equiv Z_r a(t)$$

Solve

$$\dot{\tilde{x}} = A\tilde{x} + Bu, \; \tilde{y} = C\tilde{x} + Du, \; \text{s.t.} \left(\dot{\tilde{x}} - A\tilde{x} - Bu\right) \perp_M Z_r$$

$$\Leftrightarrow \dot{a} = \underbrace{(Z_r^* MAZ_r)}_{\hat{A}} a + \underbrace{(Z_r^* MB)}_{\hat{B}} u, \; \tilde{y} = \underbrace{(CZ_r)}_{\hat{C}} a + Du$$

Proper Orthogonal Decomposition (POD)

$$\tilde{x}(t) := \sum_{j=1}^{r} a_j(t) z^{(j)} \equiv Z_r a(t)$$

Solve

$$\dot{\tilde{x}} = A\tilde{x} + Bu,\ \tilde{y} = C\tilde{x} + Du,\ \text{s.t.} \left(\dot{\tilde{x}} - A\tilde{x} - Bu\right) \perp_M Z_r$$

$$\Leftrightarrow \dot{a} = \underbrace{(Z_r^* MAZ_r)}_{\hat{A}} a + \underbrace{(Z_r^* MB)}_{\hat{B}} u,\ \tilde{y} = \underbrace{(CZ_r)}_{\hat{C}} a + Du$$

$$\boxed{\dot{a} = \hat{A}a + \hat{B}u,\ \tilde{y} = \hat{C}a + Du}$$

$\rightarrow$ solve reduced-order dynamical system for the Fourier coefficients $a_j(t)$.

Example

Heat equation

$$T_t = \kappa \Delta T \text{ in } \Omega = [0, 1]^2$$

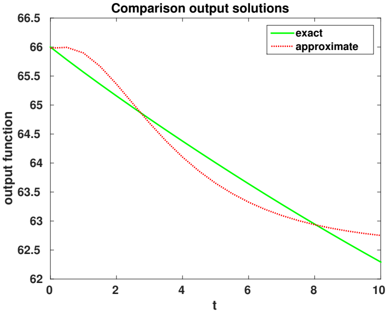
$N = 100$ grid points in x-, y-direction $t_e = 10$, $M = 20$ time steps (snapshots)
SVD, $\sigma_1, \dots, \sigma_r \geqslant \tau \|X\|$
$\tau = 10^{-2} \Rightarrow r = 3$, $\tau = 10^{-4} \Rightarrow r = 6$,

Example

Heat equation

$$T_t = \kappa \Delta T \text{ in } \Omega = [0, 1]^2$$

$N = 100$ grid points in x-, y-direction $t_e = 10$, $M = 20$ time steps (snapshots)
SVD, $\sigma_1, \ldots, \sigma_r \geqslant \tau \|X\|$
$\tau = 10^{-2} \Rightarrow r = 3$, $\tau = 10^{-4} \Rightarrow r = 6$,

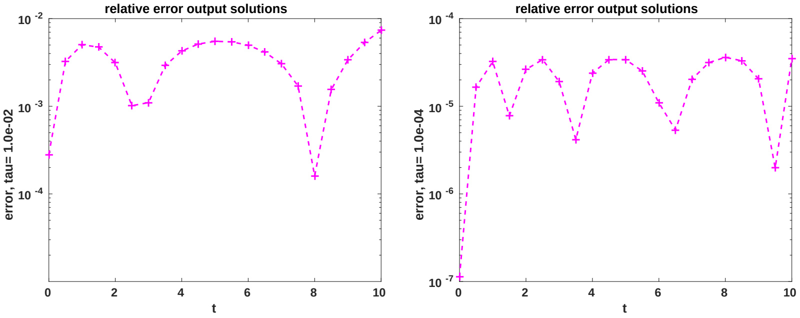


Example

Heat equation

$$T_t = \kappa \Delta T \text{ in } \Omega = [0, 1]^2$$

$N = 100$ grid points in x-, y-direction $t_e = 10$, $M = 20$ time steps (snapshots)
SVD, $\sigma_1, \dots, \sigma_r \geqslant \tau \|X\|$
$\tau = 10^{-2} \Rightarrow r = 3$, $\tau = 10^{-4} \Rightarrow r = 6$,



Remarks POD

- initial phase (offline phase) is expensive (computing snapshots, SVD)
- solving reduced order is almost for free (online phase)
- POD works well for problems like the heat equation (singular values of the analytic solution decay quadratically w.r.t. t)

$$T(x,y,t) \sim \sum_{l,m} \mu_{l,m} e^{-\kappa(l^2+m^2)\pi^2 t} \sin(l\pi x)\sin(m\pi y)$$

- POD leads to significantly higher rank for wave equations (singular values of the analytic solution decay linearly w.r.t. t)
- zero eigenvalues of the operator A on the imaginary axis (Maxwell, nullspace of the curl operator) severly interfere with POD.

Remarks POD

Variants of POD, e.g. affine-linear subspace

$$\underline{x} = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}, \quad D := X - \underline{x}\begin{pmatrix} 1 & \cdots & 1 \end{pmatrix}$$

Compute SVD of $M^{1/2}D = U\Sigma V^*$ and use for POD affine-linear model

$$\tilde{x} = \underline{x} + M^{-1/2}U_r a$$

Remarks POD

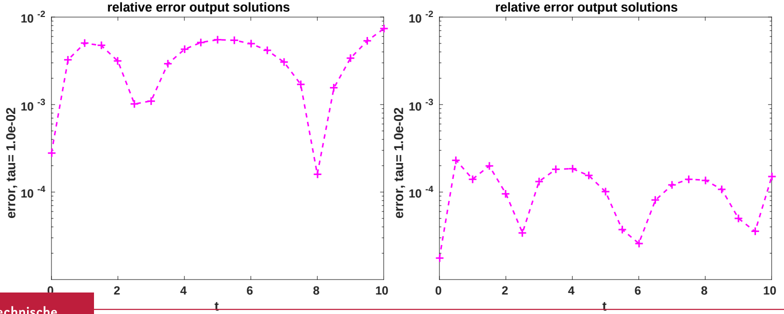
Variants of POD, e.g. affine-linear subspace

$$\underline{x} = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}, \quad D := X - \underline{x}\left(\begin{array}{ccc} 1 & \cdots & 1 \end{array} \right)$$

Compute SVD of $M^{1/2}D = U\Sigma V^*$ and use for POD affine-linear model

$$\tilde{x} = \underline{x} + M^{-1/2}U_r a$$

linear POD $\tau = 10^{-2}$, $r = 3$ versus affine-linear POD $\tau = 10^{-2}$, $r = 4$



Remarks POD

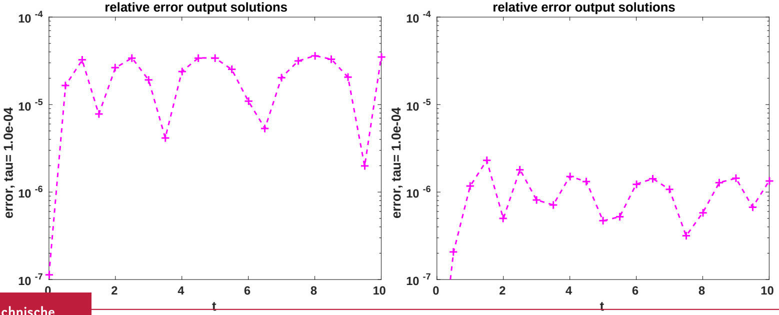
Variants of POD, e.g. affine-linear subspace

$$\underline{x} = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}, \quad D := X - \underline{x}\begin{pmatrix} 1 & \cdots & 1 \end{pmatrix}$$

Compute SVD of $M^{1/2}D = U\Sigma V^*$ and use for POD affine-linear model

$$\tilde{x} = \underline{x} + M^{-1/2}U_r a$$

linear POD $\tau = 10^{-4}$, $r = 6$ versus affine-linear POD $\tau = 10^{-4}$, $r = 7$



Outline

- Model Order Reduction
- Proper Orthogonal Decomposition
- **Balanced Truncation**
- Moment Matching
- Conclusions

Balanced Truncation (BT)

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx + Du, \end{aligned}$$

such that $\sigma(A) \subset \mathbb{C}^-$

Balanced Truncation (BT)

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx + Du, \end{aligned}$$

such that $\sigma(A) \subset \mathbb{C}^-$

$$P(t,s) = \int_t^s e^{A\tau} BB^* e^{A^*\tau} \, d\tau \text{ controllability Gramian}$$

$$Q(t,s) = \int_t^s e^{A^*\tau} C^* C e^{A\tau} \, d\tau \text{ observability Gramian}$$

$$P \equiv P(0,\infty), \;\; Q \equiv Q(0,\infty).$$

Balanced Truncation (BT)

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx + Du, \end{aligned}$$

such that $\sigma(A) \subset \mathbb{C}^-$

$$P(t,s) = \int_t^s e^{A\tau} BB^* e^{A^*\tau} \, d\tau \text{ controllability Gramian}$$

$$Q(t,s) = \int_t^s e^{A^*\tau} C^* C e^{A\tau} \, d\tau \text{ observability Gramian}$$

$$P \equiv P(0,\infty), \; Q \equiv Q(0,\infty).$$

Positive definiteness of P, Q refer to controllability/observability of the associated dynamical system.

Balanced Truncation (BT)

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx + Du, \end{aligned}$$

such that $\sigma(A) \subset \mathbb{C}^-$

$$P(t,s) = \int_t^s e^{A\tau} BB^* e^{A^*\tau} \, d\tau \text{ controllability Gramian}$$

$$Q(t,s) = \int_t^s e^{A^*\tau} C^* C e^{A\tau} \, d\tau \text{ observability Gramian}$$

$$P \equiv P(0,\infty), \;\; Q \equiv Q(0,\infty).$$

Positive definiteness of P, Q refer to controllability/observability of the associated dynamical system.
The square roots of the eigenvalues of PQ are called Hankel singular values of the dynamical system

Balanced Truncation

Remark. Hankel singular values are invariant to system transformation

Balanced Truncation

Remark. Hankel singular values are invariant to system transformation

$$\begin{array}{rcl} \dot{x} & = & Ax + Bu, \\ y & = & Cx + Du, \end{array} \quad \Longleftrightarrow \quad \begin{array}{rcl} (T^{-1}\dot{x}) & = & (T^{-1}AT)(T^{-1}x) + (T^{-1}B)u, \\ y & = & (CT)(T^{-1}x) + Du, \end{array}$$

Balanced Truncation

Remark. Hankel singular values are invariant to system transformation

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx + Du, \end{aligned} \quad \Longleftrightarrow \quad \begin{aligned} (T^{-1}\dot{x}) &= (T^{-1}AT)(T^{-1}x) + (T^{-1}B)u, \\ y &= (CT)(T^{-1}x) + Du, \end{aligned}$$

$$\equiv \quad \begin{aligned} \dot{\hat{x}} &= \hat{A}\hat{x} + \hat{B}u, \\ y &= \hat{C}\hat{x} + Du, \end{aligned}$$

Balanced Truncation

Remark. Hankel singular values are invariant to system transformation

$$\begin{array}{rcl} \dot{x} & = & Ax + Bu, \\ y & = & Cx + Du, \end{array} \iff \begin{array}{rcl} (T^{-1}\dot{x}) & = & (T^{-1}AT)(T^{-1}x) + (T^{-1}B)u, \\ y & = & (CT)(T^{-1}x) + Du, \end{array}$$

$$\equiv \begin{array}{rcl} \dot{\hat{x}} & = & \hat{A}\hat{x} + \hat{B}u, \\ y & = & \hat{C}\hat{x} + Du, \end{array}$$

$$\Rightarrow T^{-1}PT^{-*} = \hat{P},\ T^{*}QT = \hat{Q},\ \ T^{-1}PQT = \hat{P}\hat{Q}$$

Balanced Truncation

Remark. Hankel singular values are invariant to system transformation

$$\begin{array}{rcl} \dot{x} & = & Ax + Bu, \\ y & = & Cx + Du, \end{array} \iff \begin{array}{rcl} (T^{-1}\dot{x}) & = & (T^{-1}AT)(T^{-1}x) + (T^{-1}B)u, \\ y & = & (CT)(T^{-1}x) + Du, \end{array}$$

$$\equiv \begin{array}{rcl} \dot{\hat{x}} & = & \hat{A}\hat{x} + \hat{B}u, \\ y & = & \hat{C}\hat{x} + Du, \end{array}$$

$$\Rightarrow T^{-1}PT^{-*} = \hat{P},\ T^{*}QT = \hat{Q},\ \ T^{-1}PQT = \hat{P}\hat{Q}$$

A realization with (A, B, C, D) is called balanced if the ass. P and Q are diagonal.

Balanced Truncation

Remark. Hankel singular values are invariant to system transformation

$$\begin{array}{rcl} \dot{x} &=& Ax + Bu, \\ y &=& Cx + Du, \end{array} \iff \begin{array}{rcl} (T^{-1}\dot{x}) &=& (T^{-1}AT)(T^{-1}x) + (T^{-1}B)u, \\ y &=& (CT)(T^{-1}x) + Du, \end{array}$$

$$\equiv \begin{array}{rcl} \dot{\hat{x}} &=& \hat{A}\hat{x} + \hat{B}u, \\ y &=& \hat{C}\hat{x} + Du, \end{array}$$

$$\Rightarrow T^{-1}PT^{-*} = \hat{P},\ T^{*}QT = \hat{Q},\ \ T^{-1}PQT = \hat{P}\hat{Q}$$

A realization with (A, B, C, D) is called <u>balanced</u> if the ass. P and Q are diagonal.

We can construct T such that (A, B, C, D) is balanced!

Balancing the Dynamical System

1. Compute P, Q from the associated Lyapunov linear matrix equations

$$AP + PA^* + BB^* = 0, \quad A^*Q + QA + C^*C = 0.$$

Balancing the Dynamical System

1. Compute P, Q from the associated Lyapunov linear matrix equations

$$AP + PA^* + BB^* = 0, \quad A^*Q + QA + C^*C = 0.$$

2. Factorize P and Q such that

$$LL^* \stackrel{!}{=} P, \quad R^*R \stackrel{!}{=} Q.$$

Balancing the Dynamical System

1. Compute P, Q from the associated Lyapunov linear matrix equations
$$AP + PA^* + BB^* = 0, \quad A^*Q + QA + C^*C = 0.$$

2. Factorize P and Q such that
$$LL^* \stackrel{!}{=} P, \quad R^*R \stackrel{!}{=} Q.$$

3. Build $G = L^*R^*$ and compute its SVD via $G = U\Sigma V^*$

Balancing the Dynamical System

1. Compute P, Q from the associated Lyapunov linear matrix equations
$$AP + PA^* + BB^* = 0, \quad A^*Q + QA + C^*C = 0.$$

2. Factorize P and Q such that
$$LL^* \stackrel{!}{=} P, \quad R^*R \stackrel{!}{=} Q.$$

3. Build $G = L^*R^*$ and compute its SVD via $G = U\Sigma V^*$

4. Then $T := LU\Sigma^{-1/2}$ balances $(A, B, C, D) \to (\hat{A}, \hat{B}, \hat{C}, D)$

Balancing the Dynamical System

1. Compute P, Q from the associated Lyapunov linear matrix equations
$$AP + PA^* + BB^* = 0, \quad A^*Q + QA + C^*C = 0.$$

2. Factorize P and Q such that
$$LL^* \stackrel{!}{=} P, \quad R^*R \stackrel{!}{=} Q.$$

3. Build $G = L^*R^*$ and compute its SVD via $G = U\Sigma V^*$

4. Then $T := LU\Sigma^{-1/2}$ balances $(A, B, C, D) \to (\hat{A}, \hat{B}, \hat{C}, D)$

$$T^*QT = \Sigma^{-1/2}U^*L^*(R^*R)(LU\Sigma^{-1/2}) = \Sigma^{-1/2}U^*(U\Sigma V^*)(V\Sigma U^*)U\Sigma^{-1/2}) = \Sigma$$

Balancing the Dynamical System

1. Compute P, Q from the associated Lyapunov linear matrix equations

$$AP + PA^* + BB^* = 0, \quad A^*Q + QA + C^*C = 0.$$

2. Factorize P and Q such that

$$LL^* \stackrel{!}{=} P, \quad R^*R \stackrel{!}{=} Q.$$

3. Build $G = L^*R^*$ and compute its SVD via $G = U\Sigma V^*$

4. Then $T := LU\Sigma^{-1/2}$ balances $(A, B, C, D) \to (\hat{A}, \hat{B}, \hat{C}, D)$

$$T^*QT = \Sigma^{-1/2}U^*L^*(R^*R)(LU\Sigma^{-1/2}) = \Sigma^{-1/2}U^*(U\Sigma V^*)(V\Sigma U^*)U\Sigma^{-1/2}) = \Sigma$$

$$T^{-1}PT^{-*} = \Sigma^{1/2}U^*L^{-1}(LL^*)(L^{-*}U\Sigma^{1/2}) = \Sigma$$

Balancing the Dynamical System

1. Compute P, Q from the associated Lyapunov linear matrix equations
$$AP + PA^* + BB^* = 0, \quad A^*Q + QA + C^*C = 0.$$

2. Factorize P and Q such that
$$LL^* \stackrel{!}{=} P, \quad R^*R \stackrel{!}{=} Q.$$

3. Build $G = L^*R^*$ and compute its SVD via $G = U\Sigma V^*$

4. Then $T := LU\Sigma^{-1/2}$ balances $(A, B, C, D) \to (\hat{A}, \hat{B}, \hat{C}, D)$

$$T^*QT = \Sigma^{-1/2}U^*L^*(R^*R)(LU\Sigma^{-1/2}) = \Sigma^{-1/2}U^*(U\Sigma V^*)(V\Sigma U^*)U\Sigma^{-1/2}) = \Sigma$$

$$T^{-1}PT^{-*} = \Sigma^{1/2}U^*L^{-1}(LL^*)(L^{-*}U\Sigma^{1/2}) = \Sigma$$

$$\Rightarrow T^{-1}PQT = \Sigma^2$$

Benefit of Balancing the Dynamical System

How can we use the balancing decomposition for MOR?

Benefit of Balancing the Dynamical System

How can we use the balancing decomposition for MOR?

Partition the factors of the SVD $U\Sigma V^* = G$ as

$$U = [U_1, U_2],\ \Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix},\ V = [V_1, V_2]$$

(leading r columns/rows)

Benefit of Balancing the Dynamical System

How can we use the balancing decomposition for MOR?

Partition the factors of the SVD $U\Sigma V^* = G$ as

$$U = [U_1, U_2], \; \Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix}, \; V = [V_1, V_2]$$

(leading r columns/rows)

$$T_r := LU_1\Sigma_1^{-1/2}, \quad W_r := R^* V_1 \Sigma_1^{-1/2}$$

Technische Universität Braunschweig

Benefit of Balancing the Dynamical System

How can we use the balancing decomposition for MOR?

Partition the factors of the SVD $U\Sigma V^* = G$ as

$$U = [U_1, U_2],\ \Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix},\ V = [V_1, V_2]$$

(leading r columns/rows)

$$T_r := LU_1\Sigma_1^{-1/2},\quad W_r := R^* V_1\Sigma_1^{-1/2} \ \Rightarrow\ W_r^* T_r = I_r$$

(T_r, W_r are the leading r columns/rows of T, T^{-1})

Benefit of Balancing the Dynamical System

How can we use the balancing decomposition for MOR?

Partition the factors of the SVD $U\Sigma V^* = G$ as

$$U = [U_1, U_2],\ \Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix},\ V = [V_1, V_2]$$

(leading r columns/rows)

$$T_r := LU_1\Sigma_1^{-1/2},\quad W_r := R^* V_1\Sigma_1^{-1/2} \ \Rightarrow\ W_r^* T_r = I_r$$

(T_r, W_r are the leading r columns/rows of T, T^{-1})

Associated reduced order model:

$$\hat{A} := W_r^* A T_r,\ \hat{B} := W_r^* B,\ \hat{C} := C T_r,\ \hat{D} := D.$$

Benefit of Balancing the Dynamical System

How can we use the balancing decomposition for MOR?

Partition the factors of the SVD $U\Sigma V^* = G$ as

$$U = [U_1, U_2],\ \Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix},\ V = [V_1, V_2]$$

(leading r columns/rows)

$$T_r := LU_1\Sigma_1^{-1/2},\quad W_r := R^* V_1 \Sigma_1^{-1/2} \ \Rightarrow\ W_r^* T_r = I_r$$

(T_r, W_r are the leading r columns/rows of T, T^{-1})

Associated reduced order model:

$$\hat{A} := W_r^* A T_r,\ \hat{B} := W_r^* B,\ \hat{C} := C T_r,\ \hat{D} := D.$$

Error bound $\|\hat{y} - y\|_{L_2} \leqslant \|\hat{\mathcal{H}} - \mathcal{H}\|_{H_\infty} \|u\|_{L_2} = 2(\sigma_{r+1} + \cdots + \sigma_n)\|u\|_{L_2}$

Balanced Truncation Algorithm

1. Compute a low-rank approximation $L_r L_r^* \approx P$ directly from

$$AL_r L_r^* + L_r L_r^* A^* + BB^* \approx 0 = AP + PA^* + BB^*.$$

Balanced Truncation Algorithm

1. Compute a low-rank approximation $L_r L_r^* \approx P$ directly from

$$AL_rL_r^* + L_rL_r^*A^* + BB^* \approx 0 = AP + PA^* + BB^*.$$

2. Compute a low-rank approximation $R_r^* R_r \approx Q$ directly from

$$A^*R_r^*R_r + R_r^*R_rA + C^*C \approx 0 = A^*Q + QA + C^*C.$$

Balanced Truncation Algorithm

1. Compute a low-rank approximation $L_r L_r^* \approx P$ directly from
$$AL_r L_r^* + L_r L_r^* A^* + BB^* \approx 0 = AP + PA^* + BB^*.$$

2. Compute a low-rank approximation $R_r^* R_r \approx Q$ directly from
$$A^* R_r^* R_r + R_r^* R_r A + C^* C \approx 0 = A^* Q + QA + C^* C.$$

3. Set $G_r := L_r^* R_r^*$ and compute SVD of $G_r = U_r \Sigma_r V_r^*$.

Balanced Truncation Algorithm

1. Compute a low-rank approximation $L_r L_r^* \approx P$ directly from

$$AL_r L_r^* + L_r L_r^* A^* + BB^* \approx 0 = AP + PA^* + BB^*.$$

2. Compute a low-rank approximation $R_r^* R_r \approx Q$ directly from

$$A^* R_r^* R_r + R_r^* R_r A + C^* C \approx 0 = A^* Q + QA + C^* C.$$

3. Set $G_r := L_r^* R_r^*$ and compute SVD of $G_r = U_r \Sigma_r V_r^*$.

4. Set $T_r := L_r U_r \Sigma_r^{-1/2}$, $W_r := R_r^* V_r \Sigma_r^{-1/2}$

Balanced Truncation Algorithm

1. Compute a low-rank approximation $L_r L_r^* \approx P$ directly from
$$AL_rL_r^* + L_rL_r^*A^* + BB^* \approx 0 = AP + PA^* + BB^*.$$

2. Compute a low-rank approximation $R_r^* R_r \approx Q$ directly from
$$A^*R_r^*R_r + R_r^*R_rA + C^*C \approx 0 = A^*Q + QA + C^*C.$$

3. Set $G_r := L_r^* R_r^*$ and compute SVD of $G_r = U_r \Sigma_r V_r^*$.

4. Set $T_r := L_r U_r \Sigma_r^{-1/2}$, $W_r := R_r^* V_r \Sigma_r^{-1/2}$

5. Define associated reduced order model:
$$\hat{A} := W_r^* A T_r,\ \hat{B} := W_r^* B,\ \hat{C} := C T_r,\ \hat{D} := D.$$

The ADI Method

Objective: Solve $AP + PA^* + BB^* = 0$

Alternating direction implicit method (ADI): Given some P_{j-1} and shift τ_j, compute

$$(\tau_j I + A) P_{j-\frac{1}{2}} \stackrel{!}{=} -BB^* + P_{j-1}(\tau_j I - A)^*$$

$$P_j(\tau_j I + A)^* \stackrel{!}{=} -BB^* + (\tau_j I - A) P_{j-\frac{1}{2}}$$

The ADI Method

Objective: Solve $AP + PA^* + BB^* = 0$

Alternating direction implicit method (ADI): Given some P_{j-1} and shift τ_j, compute

$$(\tau_j I + A) P_{j-\frac{1}{2}} \stackrel{!}{=} -BB^* + P_{j-1}(\tau_j I - A)^*$$

$$P_j(\tau_j I + A)^* \stackrel{!}{=} -BB^* + (\tau_j I - A) P_{j-\frac{1}{2}}$$

- Convergence well-understood in the SPD case, there optimal shifts are known
- General case more complicate
- Here we want to exploit $P_j = L_j L_j^*$ explicitly
  $\longrightarrow$ low-rank Smith method, low-rank Cholesky-factor ADI

Low-Rank Cholesky-Factor ADI Method

Computing an approximate low-rank solution $L_r L_r^*$ of $AL_r L_r^* + L_r L_r^* A^* + BB^* \approx 0$

Low-Rank Cholesky-Factor ADI Method

Computing an approximate low-rank solution $L_r L_r^*$ of $AL_r L_r^* + L_r L_r^* A^* + BB^* \approx 0$

1. Compute suitable shift parameters $\tau_1, \ldots, \tau_t \in \mathbb{C}^-$

Low-Rank Cholesky-Factor ADI Method

Computing an approximate low-rank solution $L_r L_r^*$ of $AL_r L_r^* + L_r L_r^* A^* + BB^* \approx 0$

1. Compute suitable shift parameters $\tau_1, \dots, \tau_t \in \mathbb{C}^-$
2. $Z_1 := \sqrt{-2Re(\tau_1)}(\tau_1 I + A)^{-1} B$
   $L_1 := Z_1$

Low-Rank Cholesky-Factor ADI Method

Computing an approximate low-rank solution $L_r L_r^*$ of $AL_r L_r^* + L_r L_r^* A^* + BB^* \approx 0$

1. Compute suitable shift parameters $\tau_1, \ldots, \tau_t \in \mathbb{C}^-$
2. $Z_1 := \sqrt{-2Re(\tau_1)}(\tau_1 I + A)^{-1} B$
   $L_1 := Z_1$
3. For $i = 2, 3, \ldots, t$

$$
\begin{aligned}
Z_i &:= \frac{\sqrt{-2\tau_i}}{\sqrt{-2\tau_{i-1}}} \left[Z_{i-1} - (\tau_i + \bar{\tau}_{i-1}) (\tau_i I + A)^{-1} Z_{i-1} \right] \\
L_i &:= [L_{i-1}, Z_i]
\end{aligned}
$$

Low-Rank Cholesky-Factor ADI Method

Computing an approximate low-rank solution $L_r L_r^*$ of $AL_r L_r^* + L_r L_r^* A^* + BB^* \approx 0$

1. Compute suitable shift parameters $\tau_1, \ldots, \tau_t \in \mathbb{C}^-$
2. $Z_1 := \sqrt{-2Re(\tau_1)}(\tau_1 I + A)^{-1} B$
   $L_1 := Z_1$
3. For $i = 2, 3, \ldots, t$

$$\begin{aligned} Z_i &:= \frac{\sqrt{-2\tau_i}}{\sqrt{-2\tau_{i-1}}} \left[Z_{i-1} - (\tau_i + \bar{\tau}_{i-1}) (\tau_i I + A)^{-1} Z_{i-1} \right] \\ L_i &:= [L_{i-1}, Z_i] \end{aligned}$$

4. Repeat step 3 for $i = 1, 2, 3, \ldots, t$ using $\tau_0 = \tau_t$, $Z_0 = Z_t$, $L_0 = L_t$ until the “error” is small enough

Low-Rank Cholesky-Factor ADI Method

Computing an approximate low-rank solution $L_r L_r^*$ of $AL_r L_r^* + L_r L_r^* A^* + BB^* \approx 0$

1. Compute suitable shift parameters $\tau_1, \dots, \tau_t \in \mathbb{C}^-$
2. $Z_1 := \sqrt{-2Re(\tau_1)}(\tau_1 I + A)^{-1} B$
   $L_1 := Z_1$
3. For $i = 2, 3, \dots, t$

$$\begin{aligned} Z_i &:= \frac{\sqrt{-2\tau_i}}{\sqrt{-2\tau_{i-1}}} \left[Z_{i-1} - (\tau_i + \bar{\tau}_{i-1}) (\tau_i I + A)^{-1} Z_{i-1} \right] \\ L_i &:= [L_{i-1}, Z_i] \end{aligned}$$

4. Repeat step 3 for $i = 1, 2, 3, \dots, t$ using $\tau_0 = \tau_t$, $Z_0 = Z_t$, $L_0 = L_t$ until the “error” is small enough

Remark. Shift parameters are essential, optimal values only known if $-A$ is s.p.d.
“error” usually measured by changes from $L_{i-1} \to L_i$.
each update i increases $L_{i-1} \to L_i$ by rank$B$

Example Balanced Truncation

Heat equation

$$T_t = \kappa \Delta T \text{ in } \Omega = [0, 1]^2$$

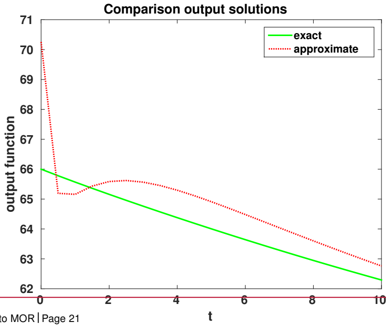
$N = 100$ grid points in x-, y-direction $t_e = 10$, $M = 20$ time steps
Low-Rank Cholesky-Factor ADI using 0.1τ
$\tau = 10^{-2} \Rightarrow 23/13$ ADI steps, $\tau = 10^{-4} \Rightarrow 40/29$ ADI steps
SVD of $G = L_r^* R_r^*$, $\sigma_1, \dots, \sigma_r \geqslant \tau \|X\|$
$\tau = 10^{-2} \Rightarrow r = 3$, $\tau = 10^{-4} \Rightarrow r = 7$,

Example Balanced Truncation

Heat equation

$$T_t = \kappa \Delta T \text{ in } \Omega = [0,1]^2$$

$N = 100$ grid points in x-, y-direction $t_e = 10$, $M = 20$ time steps
Low-Rank Cholesky-Factor ADI using 0.1τ
$\tau = 10^{-2} \Rightarrow 23/13$ ADI steps, $\tau = 10^{-4} \Rightarrow 40/29$ ADI steps
SVD of $G = L_r^* R_r^*$, $\sigma_1, \ldots, \sigma_r \geqslant \tau \|X\|$
$\tau = 10^{-2} \Rightarrow r = 3$, $\tau = 10^{-4} \Rightarrow r = 7$,

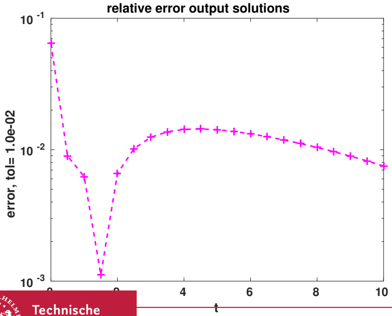


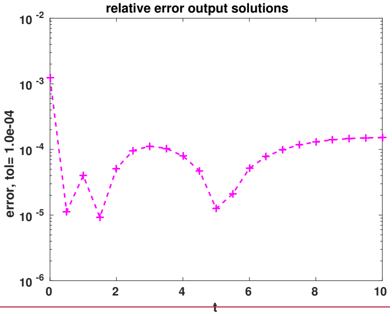
Example Balanced Truncation

Heat equation

$$T_t = \kappa \Delta T \text{ in } \Omega = [0, 1]^2$$

$N = 100$ grid points in x-, y-direction $t_e = 10$, $M = 20$ time steps
Low-Rank Cholesky-Factor ADI using 0.1τ
$\tau = 10^{-2} \Rightarrow 23/13$ ADI steps, $\tau = 10^{-4} \Rightarrow 40/29$ ADI steps
SVD of $G = L_r^* R_r^*$, $\sigma_1, \ldots, \sigma_r \geqslant \tau \|X\|$
$\tau = 10^{-2} \Rightarrow r = 3$, $\tau = 10^{-4} \Rightarrow r = 7$,





Remarks Balanced Truncation

- BT yields error bounds!
- $\sigma(A) \in \mathbb{C}^-$ essential (similar situation as for POD)
- BT can be generalized to descriptor systems $E\dot{x} = Ax + Bu$
- Additional properties such as passivity can be preserved by modifying BT (passivity: $\mathcal{H}(s)$ is analytic, $\mathcal{H}(s) + \mathcal{H}(\bar{s})^*$ is positive semidefinite f.a. $s \in \mathbb{C}^+$)

Outline

- Model Order Reduction
- Proper Orthogonal Decomposition
- Balanced Truncation
- **Moment Matching**
- Conclusions

Moment Matching-Based Model Order Reduction

Dynamical system

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx + Du, \end{aligned}$$

Transfer function

$$\mathcal{H}(s) = C(sI - A)^{-1}B$$

Moment Matching-Based Model Order Reduction

Dynamical system

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx + Du, \end{aligned}$$

Transfer function

$$\mathcal{H}(s) = C(sI - A)^{-1}B$$

Taylor expansion at s_0:

$$\mathcal{H}(s) = \sum_{p=0}^{\infty} Z_0^{(p)}(s - s_0)^p, \text{ where } Z_0^{(p)} = -C(A - s_0 I)^{-p-1}B$$

Technische Universität Braunschweig

Moment Matching-Based Model Order Reduction

Dynamical system

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx + Du, \end{aligned}$$

Transfer function

$$\mathcal{H}(s) = C(sI - A)^{-1}B$$

Taylor expansion at s_0:

$$\mathcal{H}(s) = \sum_{p=0}^{\infty} Z_0^{(p)} (s - s_0)^p, \text{ where } Z_0^{(p)} = -C(A - s_0 I)^{-p-1}B$$

$X_0^{(p)} := -(A - s_0 I)^{-p-1}B$ input moments, Taylor coefficients $Z_0^{(p)}$ output moments.
$Y_0^{(p)} := -C(A - s_0 I)^{-p-1}$

Elementary Considerations

Suppose that A is simple, $T^{-1}AT = \Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_n)$, $Re(\lambda_j) < 0$.

Elementary Considerations

Suppose that A is simple, $T^{-1}AT = \Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$, $Re(\lambda_j) < 0$.

Partition $T = [T_r, \tilde{T}]$, $T^{-1} = \begin{bmatrix} W_r^* \\ \tilde{W}^* \end{bmatrix}$.

Elementary Considerations

Suppose that A is simple, $T^{-1}AT = \Lambda = \mathrm{diag}\,(\lambda_1, \ldots, \lambda_n)$, $Re(\lambda_j) < 0$.

Partition $T = [T_r, \tilde{T}]$, $T^{-1} = \begin{bmatrix} W_r^* \\ \tilde{W}^* \end{bmatrix}$.

associated reduced-order system

$$\hat{A} := W_r^* A T_r,\ \hat{B} := W_r^* B,\ \hat{C} := C T_r,\ \hat{D} := D$$

$$\|\hat{y} - y\|_{L_2} \leqslant \|\hat{\mathcal{H}} - \mathcal{H}\|_{H_\infty} \|u\|_{L_2} \leqslant \|C\|_2 \|B\|_2 \mathrm{cond}_2(T) \max_{j>r} \frac{1}{|Re(\lambda_j)|} \|u\|_{L_2}$$

Elementary Considerations

Suppose that A is simple, $T^{-1}AT = \Lambda = \mathrm{diag}\,(\lambda_1, \ldots, \lambda_n)$, $Re(\lambda_j) < 0$.

Partition $T = [T_r, \tilde{T}]$, $T^{-1} = \begin{bmatrix} W_r^* \\ \tilde{W}^* \end{bmatrix}$.

associated reduced-order system

$$\hat{A} := W_r^* A T_r,\ \hat{B} := W_r^* B,\ \hat{C} := C T_r,\ \hat{D} := D$$

$$\|\hat{y} - y\|_{L_2} \leqslant \|\hat{\mathcal{H}} - \mathcal{H}\|_{H_\infty} \|u\|_{L_2} \leqslant \|C\|_2 \|B\|_2 \mathrm{cond}_2(T) \max_{j>r} \frac{1}{|Re(\lambda_j)|} \|u\|_{L_2}$$

- matching eigenvectors/eigenvalues close to the imaginary axis advantageous
- Taylor expansion of $\mathcal{H}(s)$ sensitive w.r.t. expansion point s_0
- Often applications only require $\max_{\omega \in [f_{\min}, f_{\max}]} \|\hat{\mathcal{H}}(2\pi i \omega) - \mathcal{H}(2\pi i \omega)\|_\infty$ to be small

Krylov Subspace Methods

Krylov subspace

$$\mathcal{K}_p(A, b) = \text{span}\{b, Ab, \ldots, A^{p-1}b\}$$

Krylov Subspace Methods

Krylov subspace

$$\mathcal{K}_p(A, b) = \mathrm{span}\{b, Ab, \ldots, A^{p-1}b\}$$

- input Krylov subpace

$$X_0^{(p)} = (A - s_0 I)^{-p-1} B \in \mathcal{K}_{p+1}((s_0 I - A)^{-1}, B_0), \text{ where } B_0 = (A - s_0 I)^{-1} B.$$

- output Krylov subspace

$$(Y_0^{(p)})^* = (A^* - \bar{s}_0 I)^{-p-1} C^* \in \mathcal{K}_{p+1}((s_0 I - A)^{-*}, C_0^*), \text{ where } C_0^* = (A - s_0 I)^{-*} C^*.$$

Krylov Subspace Methods

Krylov subspace

$$\mathcal{K}_p(A,b) = \operatorname{span}\{b, Ab, \dots, A^{p-1}b\}$$

- input Krylov subpace

$$X_0^{(p)} = (A - s_0 I)^{-p-1}B \in \mathcal{K}_{p+1}((s_0 I - A)^{-1}, B_0), \text{ where } B_0 = (A - s_0 I)^{-1}B.$$

- output Krylov subspace

$$(Y_0^{(p)})^* = (A^* - \bar{s}_0 I)^{-p-1}C^* \in \mathcal{K}_{p+1}((s_0 I - A)^{-*}, C_0^*), \text{ where } C_0^* = (A - s_0 I)^{-*}C^*.$$

Main idea of moment matching methods:
Compute T_r and/or W_r from $\mathcal{K}_r((s_0 I - A)^{-1}, B_0)$, $\mathcal{K}_r((s_0 I - A)^{-*}, C_0^*)$ s.t. $W_r^* T_r = I$

Krylov Subspace Methods

Krylov subspace

$$\mathcal{K}_p(A, b) = \text{span}\{b, Ab, \ldots, A^{p-1}b\}$$

- input Krylov subpace

$$X_0^{(p)} = (A - s_0 I)^{-p-1} B \in \mathcal{K}_{p+1}((s_0 I - A)^{-1}, B_0), \text{ where } B_0 = (A - s_0 I)^{-1} B.$$

- output Krylov subspace

$$(Y_0^{(p)})^* = (A^* - \bar{s}_0 I)^{-p-1} C^* \in \mathcal{K}_{p+1}((s_0 I - A)^{-*}, C_0^*), \text{ where } C_0^* = (A - s_0 I)^{-*} C^*.$$

Main idea of moment matching methods:
Compute T_r and/or W_r from $\mathcal{K}_r((s_0 I - A)^{-1}, B_0)$, $\mathcal{K}_r((s_0 I - A)^{-*}, C_0^*)$ s.t. $W_r^* T_r = I$

associated reduced-order system

$$\hat{A} := W_r^* A T_r, \ \hat{B} := W_r^* B, \ \hat{C} := C T_r, \ \hat{D} := D$$

associated reduced-order transfer function

$$\hat{\mathcal{H}}_r(s) := \hat{C}(sI - \hat{A})^{-1} \hat{B}$$

Moment Matching Methods

Consequences

$$\hat{\mathcal{H}}_r(s) := \sum_{p=0}^{t-1} Z_0^{(p)}(s-s_0)^p + \sum_{p=t}^{\infty} \hat{Z}_0^{(p)}(s-s_0)^p$$

Moment Matching Methods

Consequences

$$\hat{\mathcal{H}}_r(s) := \sum_{p=0}^{t-1} Z_0^{(p)}(s-s_0)^p + \sum_{p=t}^{\infty} \hat{Z}_0^{(p)}(s-s_0)^p$$

- If $T_r \in \mathcal{K}_r((s_0 I - A)^{-1}, B_0)$ and W_r s.t. $W_r^* T_r = I$, then $t \geqslant r$
- If $W_r \in \mathcal{K}_r((s_0 I - A)^{-*}, C_0^*)$ and T_r s.t. $W_r^* T_r = I$, then $t \geqslant r$
- If $T_r \in \mathcal{K}_r((s_0 I - A)^{-1}, B_0)$ and $W_r \in \mathcal{K}_r((s_0 I - A)^{-*}, C_0^*)$ $W_r^* T_r = I$, then $t \geqslant 2r$.

Krylov Subspace Methods — Lanczos (PVL)

Lanczos-type method (Padé via Lanczos)

Krylov Subspace Methods — Lanczos (PVL)

Lanczos-type method (Padé via Lanczos)

Generate dual bases T_r of $\mathcal{K}_r((s_0 I - A)^{-1}, B_0)$ <u>and</u> W_r of $\mathcal{K}_r((s_0 I - A)^{-*}, C_0^*)$

Technische Universität Braunschweig

Krylov Subspace Methods — Lanczos (PVL)

Lanczos-type method (Padé via Lanczos)

Generate dual bases T_r of $\mathcal{K}_r((s_0 I - A)^{-1}, B_0)$ and W_r of $\mathcal{K}_r((s_0 I - A)^{-*}, C_0^*)$

We obtain $(s_0 I - A)^{-1} T_{r-1} = T_r \underline{L}_r$ and $(s_0 I - A)^{-*} W_{r-1} = W_r \hat{\underline{L}}_r$, where

Krylov Subspace Methods — Lanczos (PVL)

Lanczos-type method (Padé via Lanczos)

Generate dual bases T_r of $\mathcal{K}_r((s_0 I - A)^{-1}, B_0)$ <u>and</u> W_r of $\mathcal{K}_r((s_0 I - A)^{-*}, C_0^*)$

We obtain $(s_0 I - A)^{-1} T_{r-1} = T_r \underline{L}_r$ and $(s_0 I - A)^{-*} W_{r-1} = W_r \hat{\underline{L}}_r$, where

$$W_r^* T_r = I \text{ and}$$

Krylov Subspace Methods — Lanczos (PVL)

Lanczos-type method (Padé via Lanczos)

Generate dual bases T_r of $\mathcal{K}_r((s_0 I - A)^{-1}, B_0)$ and W_r of $\mathcal{K}_r((s_0 I - A)^{-*}, C_0^*)$

We obtain $(s_0 I - A)^{-1} T_{r-1} = T_r \underline{L}_r$ and $(s_0 I - A)^{-*} W_{r-1} = W_r \underline{\hat{L}}_r$, where

$$W_r^* T_r = I \text{ and}$$

$$\underline{L}_r = \left[\begin{array}{cccc} l_{11} & l_{12} & & 0 \\ l_{21} & l_{22} & l_{23} & \\ & \ddots & \ddots & \ddots \\ 0 & & l_{r-1,r-2} & l_{r-1,r-1} \\ \hline 0 & & & l_{r,r-1} \end{array}\right], \quad \underline{\hat{L}}_r = \left[\begin{array}{cccc} \bar{l}_{11} & \bar{l}_{21} & & 0 \\ \bar{l}_{12} & \bar{l}_{22} & \bar{l}_{32} & \\ & \ddots & \ddots & \ddots \\ 0 & & \bar{l}_{r-2,r-1} & \bar{l}_{r-1,r-1} \\ \hline 0 & & & \bar{l}_{r-1,r} \end{array}\right].$$

Krylov Subspace Methods — Lanczos (PVL)

$t := (s_0 I - A)^{-1} B$, $w := (s_0 I - A)^{-*} C^*$
$T_1 := t/\|t\|$, $W_1 = w/(T_1^* w)$

$t := (s_0 I - A)^{-1} B$, $w := (s_0 I - A)^{-*} C^*$
$T_1 := t / \|t\|$, $W_1 = w / (T_1^* w)$

$l_{01} = l_{10} := 0$

Krylov Subspace Methods — Lanczos (PVL)

$t := (s_0 I - A)^{-1} B$, $w := (s_0 I - A)^{-*} C^*$
$T_1 := t / \|t\|$, $W_1 = w / (T_1^* w)$

$l_{01} = l_{10} := 0$
For $i = 1, 2, \dots, r-1$
  $t := (s_0 I - A)^{-1} T_i$
  $l_{ii} := W_i^* t$
  $t := t - T_i l_{ii} - T_{i-1} l_{i-1,i}$
  $l_{i+1,i} := \|t\|$, $T_{i+1} := t / l_{i+1,i}$

  $w := (s_0 I - A)^{-*} W_i$
  $w := w - W_i \bar{l}_{ii} - W_{i-1} \bar{l}_{i,i-1}$
  $l_{i,i+1} := w^* T_i$, $W_{i+1} := w / \bar{l}_{i,i+1}$

Krylov Subspace Methods — Arnoldi

Arnoldi-type method

Krylov Subspace Methods — Arnoldi

Arnoldi-type method

Generate orthonormal basis T_r of $\mathcal{K}_r((s_0 I - A)^{-1}, B_0)$

Krylov Subspace Methods — Arnoldi

Arnoldi-type method

Generate orthonormal basis T_r of $\mathcal{K}_r((s_0 I - A)^{-1}, B_0)$

We obtain $(s_0 I - A)^{-1} T_{r-1} = T_r \underline{H}_r$, where $T_r^* T_r = I$ and

$$\underline{H}_r = \left[\begin{array}{cccc} h_{11} & \cdots & \cdots & h_{1,r-1} \\ h_{21} & h_{22} & & \vdots \\ & \ddots & \ddots & \vdots \\ 0 & & h_{r-1,r-2} & h_{r-1,r-1} \\ \hline 0 & & & h_{r,r-1} \end{array}\right].$$

Krylov Subspace Methods — Arnoldi

$t := (s_0 I - A)^{-1} B$
$T_1 := t / \|t\|$

Krylov Subspace Methods — Arnoldi

$t := (s_0 I - A)^{-1} B$
$T_1 := t / \|t\|$

For $i = 1, 2, \ldots, r-1$
  $t := (s_0 I - A)^{-1} T_i$
  For $j = 1, 2, \ldots, i$
    $h_{ji} := T_j^* t$
    $t := t - T_j h_{ji}$
  $h_{i+1,i} := \|t\|$, $T_{i+1} := t / h_{i+1,i}$

Example Krylov Subspace Methods

Heat equation

$$T_t = \kappa \Delta T \text{ in } \Omega = [0, 1]^2$$

$N = 100$ grid points in x-, y-direction $t_e = 10$, $M = 20$ time steps
Use expansion point $s_0 = 0$
PVL use $r = 3$, $r = 5$, Arnoldi uses $r = 3, 5, 10, 15, 20$

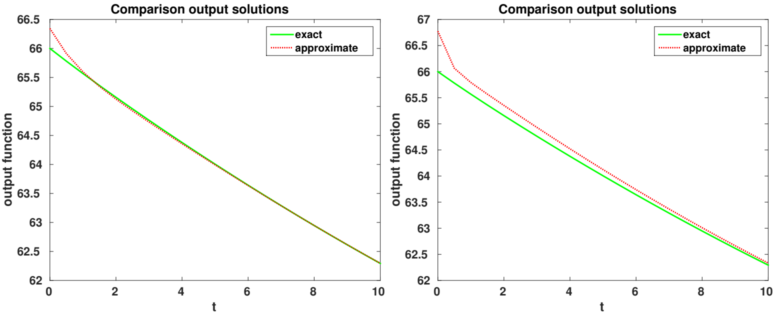
Example Krylov Subspace Methods

Heat equation

$$T_t = \kappa \Delta T \text{ in } \Omega = [0,1]^2$$

$N = 100$ grid points in x-, y-direction $t_e = 10$, $M = 20$ time steps
Use expansion point $s_0 = 0$
PVL use $r = 3$, $r = 5$, Arnoldi uses $r = 3, 5, 10, 15, 20$

PVL ($r = 3$) versus Arnoldi ($r = 3$)



Example Krylov Subspace Methods

Heat equation

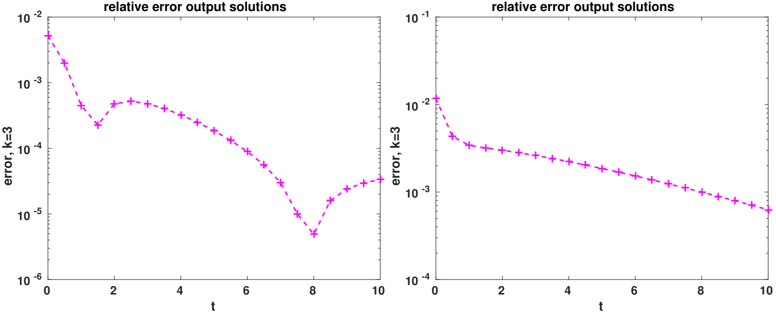
$$T_t = \kappa \Delta T \text{ in } \Omega = [0,1]^2$$

$N = 100$ grid points in x-, y-direction $t_e = 10$, $M = 20$ time steps

Use expansion point $s_0 = 0$

PVL use $r = 3$, $r = 5$, Arnoldi uses $r = 3, 5, 10, 15, 20$

PVL ($r = 3$) versus Arnoldi ($r = 3$)



Example Krylov Subspace Methods

Heat equation

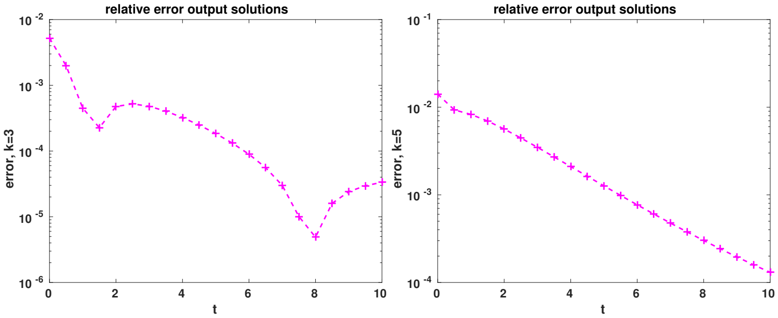
$$T_t = \kappa \Delta T \text{ in } \Omega = [0, 1]^2$$

$N = 100$ grid points in x-, y-direction $t_e = 10$, $M = 20$ time steps

Use expansion point $s_0 = 0$

PVL use $r = 3$, $r = 5$, Arnoldi uses $r = 3, 5, 10, 15, 20$

PVL ($r = 3$) versus Arnoldi ($r = 5$)



Example Krylov Subspace Methods

Heat equation

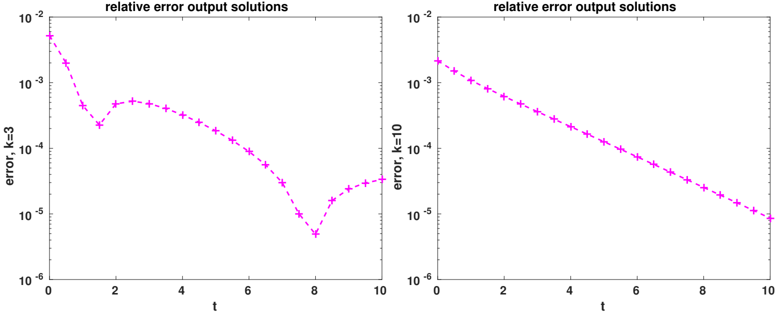
$$T_t = \kappa \Delta T \text{ in } \Omega = [0, 1]^2$$

$N = 100$ grid points in x-, y-direction $t_e = 10$, $M = 20$ time steps

Use expansion point $s_0 = 0$

PVL use $r = 3$, $r = 5$, Arnoldi uses $r = 3, 5, 10, 15, 20$

PVL ($r = 3$) versus Arnoldi ($r = 10$)



Example Krylov Subspace Methods

Heat equation

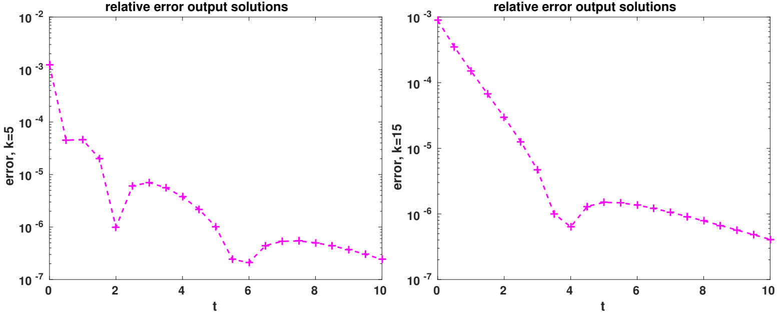
$$T_t = \kappa \Delta T \text{ in } \Omega = [0,1]^2$$

$N = 100$ grid points in x-, y-direction $t_e = 10$, $M = 20$ time steps

Use expansion point $s_0 = 0$

PVL use $r = 3$, $r = 5$, Arnoldi uses $r = 3, 5, 10, 15, 20$

PVL ($r = 5$) versus Arnoldi ($r = 15$)



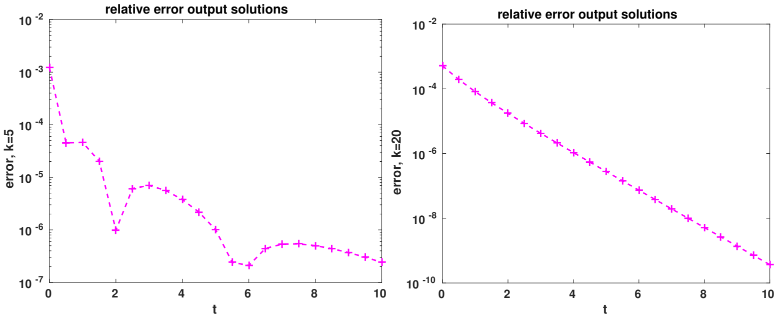
Example Krylov Subspace Methods

Heat equation

$$T_t = \kappa \Delta T \text{ in } \Omega = [0,1]^2$$

$N = 100$ grid points in x-, y-direction $t_e = 10$, $M = 20$ time steps
Use expansion point $s_0 = 0$
PVL use $r = 3$, $r = 5$, Arnoldi uses $r = 3, 5, 10, 15, 20$

PVL ($r = 5$) versus Arnoldi ($r = 20$)



Remarks Krylov Subspace Methods

- PVL matches twice as many moments, but unstable (reorthogonalization, break downs), no symmetry preservation
- Arnoldi more stable, either B or C^* is taken into acount, stable, preserves symmetry

Remarks Krylov Subspace Methods

- PVL matches twice as many moments, but unstable (reorthogonalization, break downs), no symmetry preservation
- Arnoldi more stable, either B or C^* is taken into acount, stable, preserves symmetry

- size r of the subspace not known in advance, no error bounds
- possibly multiple calls necessary, e.g. compare $\|\hat{\mathcal{H}}_{r_l}(s) - \hat{\mathcal{H}}_{r_{l+1}}(s)\| / \|\hat{\mathcal{H}}_{r_{l+1}}(s)\|$ from two subsequent calls

Remarks Krylov Subspace Methods

- PVL matches twice as many moments, but unstable (reorthogonalization, break downs), no symmetry preservation
- Arnoldi more stable, either B or C^* is taken into acount, stable, preserves symmetry

- size r of the subspace not known in advance, no error bounds
- possibly multiple calls necessary, e.g. compare $\|\hat{\mathcal{H}}_{r_l}(s) - \hat{\mathcal{H}}_{r_{l+1}}(s)\| / \|\hat{\mathcal{H}}_{r_{l+1}}(s)\|$ from two subsequent calls

- Krylov subspace methods are extremeley sensitive to the choice of s_0, location of eigenvalues of A is helpful (e.g. real and negative), frequency range as well
- shift s_0 on the imaginary axis, usually complex-valued matrices T, W

Rational Krylov Subspace Methods

- Use multiple Taylor expansions at $s_1, \ldots, s_l$

$$\mathcal{H}(s) = \sum_{p=0}^{\infty} Z_j^{(p)} (s - s_j)^p, \; j = 1, \ldots, l$$

Rational Krylov Subspace Methods

- Use multiple Taylor expansions at $s_1, \ldots, s_l$

$$\mathcal{H}(s) = \sum_{p=0}^{\infty} Z_j^{(p)} (s - s_j)^p, \; j = 1, \ldots, l$$

- Rational Krylov method: Compute basis T_r for the Krylov subspaces

$$\sum_{j=1}^{l} \mathcal{K}_{r_j}((s_j I - A)^{-1}, B_j), \text{ where } B_j = (s_j I - A)^{-1} B$$

Rational Krylov Subspace Methods

- Use multiple Taylor expansions at $s_1, \dots, s_l$

$$\mathcal{H}(s) = \sum_{p=0}^{\infty} Z_j^{(p)} (s - s_j)^p, \; j = 1, \dots, l$$

- Rational Krylov method: Compute basis T_r for the Krylov subspaces

$$\sum_{j=1}^{l} \mathcal{K}_{r_j}((s_j I - A)^{-1}, B_j), \text{ where } B_j = (s_j I - A)^{-1} B$$

and/or possibly basis W_r (such that $W_r^* T_r = I$) for the Krylov subspaces

$$\sum_{j=1}^{l} \mathcal{K}_{r_j}((s_j I - A)^{-*}, C_j^*), \text{ where } C_j = (s_j I - A)^{-*} C^*$$

$$r = r_1 + \cdots + r_l$$

Rational Arnoldi Methods

Lemma (Partial Fraction Decomposition)

Suppose that $s_i \neq s_j$, then

$$(s_i I - A)^{-1} \cdot (s_j I - A)^{-p+1} B_j \in \mathcal{K}_p((s_j I - A)^{-1}, B_j) + \mathcal{K}_1((s_i I - A)^{-1}, B_i)$$

Rational Arnoldi Methods

Lemma (Partial Fraction Decomposition)

Suppose that $s_i \neq s_j$, then

$$(s_i I - A)^{-1} \cdot (s_j I - A)^{-p+1} B_j \in \mathcal{K}_p((s_j I - A)^{-1}, B_j) + \mathcal{K}_1((s_i I - A)^{-1}, B_i)$$

- $\Rightarrow$ mixing inverses with different shifts leads to a separate sum of Krylov subspaces, no “mixed powers of inverses”
- We may run the Arnoldi method with shifts $s_1, \ldots, s_l$ simultaneously, e.g., one shift after another or cyclically

Example Rational Arnoldi Method

Heat equation

$$T_t = \kappa \Delta T \text{ in } \Omega = [0, 1]^2$$

$N = 100$ grid points in x-, y-direction $t_e = 10$, $M = 20$ time steps
Use expansion points $s_j \in \{0, \pm i, \pm 2i, \pm 3i, \ldots\}$, $j = 1, \ldots, l$
rational Arnoldi uses $r = 5, 10$, $l = 1, 3, 5$ cyclically

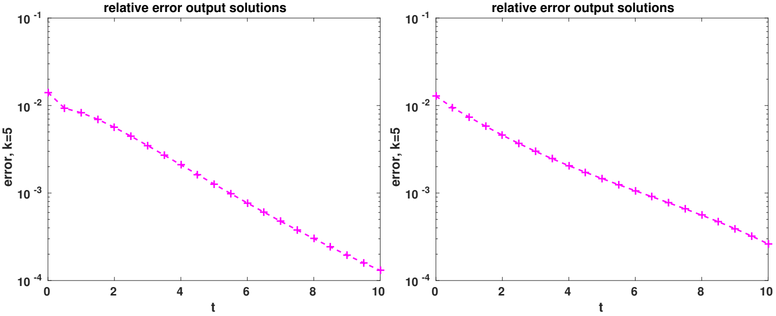
Example Rational Arnoldi Method

Heat equation

$$T_t = \kappa \Delta T \text{ in } \Omega = [0,1]^2$$

$N = 100$ grid points in x-, y-direction $t_e = 10$, $M = 20$ time steps
Use expansion points $s_j \in \{0, \pm i, \pm 2i, \pm 3i, \ldots\}$, $j = 1, \ldots, l$
rational Arnoldi uses $r = 5, 10$, $l = 1, 3, 5$ cyclically

Arnoldi ($r = 5$) versus rational Arnoldi ($r = 5$, $l = 3$)



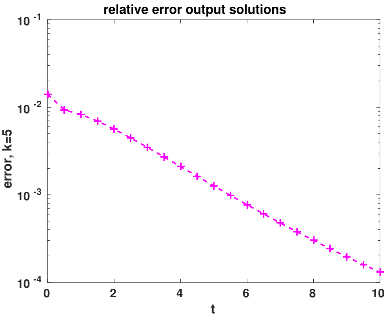
Example Rational Arnoldi Method

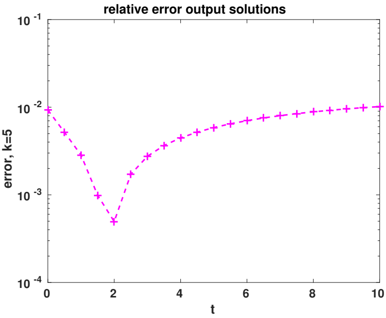
Heat equation

$$T_t = \kappa \Delta T \text{ in } \Omega = [0, 1]^2$$

$N = 100$ grid points in x-, y-direction $t_e = 10$, $M = 20$ time steps
Use expansion points $s_j \in \{0, \pm i, \pm 2i, \pm 3i, \dots\}$, $j = 1, \dots, l$
rational Arnoldi uses $r = 5, 10$, $l = 1, 3, 5$ cyclically

Arnoldi ($r = 5$) versus rational Arnoldi ($r = 5$, $l = 5$)





Example Rational Arnoldi Method

Heat equation

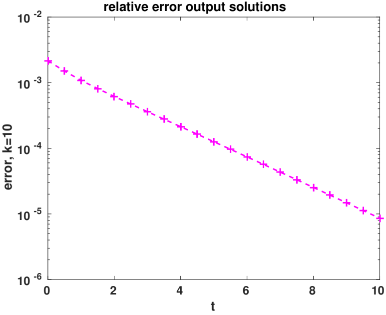
$$T_t = \kappa \Delta T \text{ in } \Omega = [0,1]^2$$

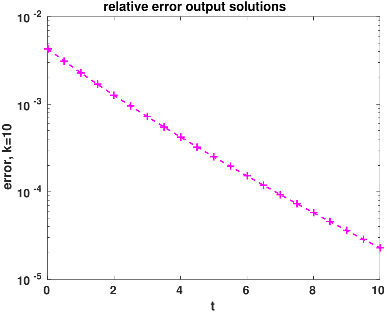
$N = 100$ grid points in x-, y-direction $t_e = 10$, $M = 20$ time steps

Use expansion points $s_j \in \{0, \pm i, \pm 2i, \pm 3i, \dots\}$, $j = 1, \dots, l$

rational Arnoldi uses $r = 5, 10$, $l = 1, 3, 5$ cyclically

Arnoldi ($r = 10$) versus rational Arnoldi ($r = 10$, $l = 3$)





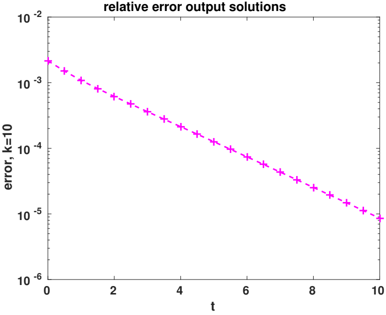
Example Rational Arnoldi Method

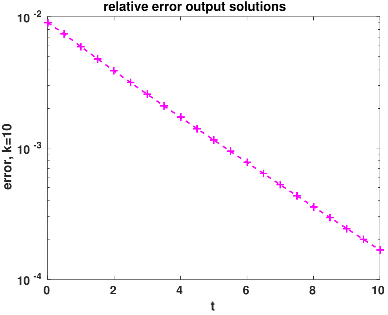
Heat equation

$$T_t = \kappa \Delta T \text{ in } \Omega = [0,1]^2$$

$N = 100$ grid points in x-, y-direction $t_e = 10$, $M = 20$ time steps
Use expansion points $s_j \in \{0, \pm i, \pm 2i, \pm 3i, \ldots\}$, $j = 1, \ldots, l$
rational Arnoldi uses $r = 5, 10$, $l = 1, 3, 5$ cyclically

Arnoldi ($r = 10$) versus rational Arnoldi ($r = 10$, $l = 5$)





Remarks Rational Arnoldi Method

- Choice of multiple shifts not clear in advance, sometimes one shift is enough
- Multiple shifts may reduce the error $\|\hat{\mathcal{H}}_{r_l}(s) - \hat{\mathcal{H}}_{r_{l+1}}(s)\| / \|\hat{\mathcal{H}}_{r_{l+1}}(s)\|$ between two subsequent rational Arnoldi calls more uniformly
- Adaptive strategies to select s_j exist
- Multiple shifts require more *LU* decompositions

Technische Universität Braunschweig

Outline

- Model Order Reduction
- Proper Orthogonal Decomposition
- Balanced Truncation
- Moment Matching
- **Conclusions**

Conclusions

- Three different approaches to perform model order reduction presented
- Discussion here only simplified!
- no clear winner, problem-dependent
- POD: use SVD of a snapshot sequence,
  BT: use low-rank approximation of the associated Gramians
  Moment Matching: build bases of the associated Krylov subspace
- Many additional topics to be discussed (generalizations to $E\dot{x} = Ax + Bu$, error estimates for Krylov-type methods, numerical solvers for solving the shifted systems, parametrized systems, time-dependent systems, nonlinear systems,. . .)